



HTML and CSS. Site layout. Best practices.

Roman Melnyk

AGENDA

Knowledge base

HTML / CSS basics

- HTML tags, their attributes
- CSS specification and possibilities
 - rules for tags, classes, pseudoclasses
 - rule dependencies
 - 'elephant vs. whale'

Best practices

- Wrapping elements
- Setting classes
- Using names
- Classes tricks

KNOWLEDGE BASE

TextText**TextText***TextText*TextTextTextText

TextText

TextText

Text

Text

Text

TextText

TextText

KNOWLEDGE BASE

Difference between:

- `.style1 .style2 {color: red;}`
- `.style1, .style2 {color: green;}`
- `.style1.style2 {color: blue;}`

KNOWLEDGE BASE

What does it mean:

- `.style1 {background: url(gfx/logo.png) -20px 0px;}`

HTML / CSS BASICS – HTML TAGS, THEIR ATTRIBUTES

```
<span id="name" class="classname">
```

Content

```
</span>
```

```
<div>, <p>, <a>, ...
```

```
<ul>, <ol>, <li>, ...
```

```
<img/>, <br/>
```

```
<table>
```

Block elements and inline elements

http://www.w3schools.com/html/html_elements.asp

http://www.w3schools.com/html/html_attributes.asp

HTML / CSS BASICS – CSS SPECIFICATION AND POSSIBILITIES

```
span {color: red;}  
.classname {color: blue;}  
a:hover {color: orange;}  
#id {color: yellow;}
```

http://www.w3schools.com/css/css_id_class.asp

HTML / CSS BASICS – CSS RULE DEPENDENCIES

```
div span {color: red;}  
    div .c1 #c2 {color: blue;}  
div>span {color: yellow;}  
#id1, #id2 {color: yellow;}  
.c1.c2 {color: yellow;}  
    .c1.c2 a:hover {color: yellow;}
```

http://www.w3schools.com/css/css_grouping_nesting.asp

HTML / CSS BASICS –

CSS RULES WEIGHT

1. !important;

```
.class {color: #454545 !important;}
```

2. Inline style; script set style

```
<span style="font-size: 18px;">
```

3. CSS definitions combination:

1. tagname = 1
2. classname = 10
3. id = 100

```
#id.class a span.i {font-style: italic;}
```

1. Browser/OS defaults

BEST PRACTICES

Use clean and clear HTML with CSS:

- avoid inline styling;
- try to avoid `<table>` tag;
- use clear names for id's and classes;

Wrap elements and functional parts in `<div>` for simple and clear styling:

- Divide et impera!
- Progressive JPEG method

BEST PRACTICES

Set classes, nevertheless it looks unnecessary

- Browser specific content; using classes instead of hacks
 - Use JS for Browser determination
 - `<!--IF IE-->` for ie.css
- Easy restyling/rebranding
- Easy access to abstractions and enhance styling possibilities

BEST PRACTICES

Use multiple classes for one elements:

- Abstract classes
`{float: left;}, {position: relative;}, {display: none;}`
 - Similar elements styling
 - Text**TextTextTextTextText** example
 - form elements event-depended styling
- Hover styling
`(a img {behavoir_1} / a:hover img {behavoir_2})`
- Menus, events, notifications, etc. – it's web 2.0, man!

REFERENCES

<http://www.w3schools.com>

<http://google.com>



Thank you!

A decorative image on the left side of the slide shows several white, three-dimensional cubes arranged on a surface. In the background, there is a blue geometric pattern, possibly a map or a design. The overall scene is brightly lit, with a soft glow.